Decision Support Systems 53 (2012) 704–711

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Feature selection for sentiment analysis based on content and syntax models

Adnan Duric, Fei Song *

School of Computer Science, University of Guelph, 50 Stone Road East Guelph, Ontario, Canada N1G 2W1

ARTICLE INFO

Available online 23 May 2012

Keywords:
Sentiment analysis
Text classification
Feature selection
Maximum entropy modeling
Topic modeling
Content and Syntax models

ABSTRACT

Recent solutions for sentiment analysis have relied on feature selection methods ranging from lexicon-based approaches where the set of features are generated by humans, to approaches that use general statistical measures where features are selected solely on empirical evidence. The advantage of statistical approaches is that they are fully automatic, however, they often fail to separate features that carry sentiment from those that do not. In this paper we propose a set of new feature selection schemes that use a Content and Syntax model to automatically learn a set of features in a review document by separating the entities that are being reviewed from the subjective expressions that describe those entities in terms of polarities. By focusing only on the subjective expressions and ignoring the entities, we can choose more salient features for document-level sentiment analysis. The results obtained from using these features in a maximum entropy classifier are competitive with the state-of-the-art machine learning approaches.

1. Introduction

In recent years, Web 2.0 has exploded with user-generated platforms such as blogs, discussion forums, and social networks. Opinionated consumers have at their disposal unprecedented reach and power by which to share their brand experiences and opinions. Major companies are now beginning to realize that their consumers wield more influences than ever before and as a result are beginning to look at reviews of their products and services more closely. In fact, this phenomenon is not only related to companies that make products and services, but also public figures and celebrities. These entities can then respond to consumer insights and general public sentiments by monitoring the places that generate them, and ultimately gain an advantage that has so far only come about with an extensive and usually expensive consumer research campaign.

According to two surveys of more than 2000 American adults [6]: 81% of Internet users have done online research on a product at least once and between 73% and 87% report that reviews had a significant influence on their purchases. However, the sheer volume of user-generated opinion content has become so enormous that most companies and public figures have to spend a lot of time engaged in it to find an overall sentiment. Indeed, even fellow consumers spend a lot of time sifting through many reviews in order to find that one review that discusses features and matters that are important to them.

One way for organizing such data is *text classification*, which involves mapping documents into *topical* categories based on the occurrences of particular features. Sentiment Analysis (SA) can be framed as a text classification task where the categories are *polarities* such as *positive* and *negative*. However, the similarities end here. Whereas general text classification is concerned with features that distinguish different topics, sentiment analysis deals with features about subjectivity, affect, emotion, and points-of-view that *describe* or *modify* the related entities. Since user-generated review documents contain both kinds of features, SA solutions ultimately face the challenge of separating the objective content from the subjective content describing it.

For example, taking a segment from a randomly chosen document in Pang et al.'s [13] movie review corpus,[1] we see how entities and modifiers are related to each other:

> … Of course, it helps that **Kaye** has an **actor** as *talented* as **Norton** to play **this part**. It's *astonishing* how *frightening* **Norton** looks with a shaved head and a swastika on his chest. … Visually, **the film** is *very powerful*. **Kaye** indulges in a lot of *interesting* **artistic choices**, and most of **them** *work nicely*.

Indeed, most of the information about an entity that relates it to a particular polarity comes from the *modifying* words. In the example above, these words are adjectives such as *talented*, *frightening*, *interesting*, and *powerful*. They can also be verbs such as *work* and adverbs such as *nicely*. The entities are represented by various nouns and pronouns such as: **Kaye**, **Norton**, **actor** and **them**.

1.1. Applications

"What other people think" has always been an important piece of information for most of us when we look to make a decision. SA is an

* Corresponding author. Tel.: +1 519 824 4120x58067; fax: +1 519 837 0323.
E-mail addresses: aduric@uoguelph.ca (A. Duric), fsong@uoguelph.ca (F. Song).

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/.

0167-9236/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2012.05.023

important research area in that it leads to useful applications in a variety of ways.

- *Online customer reviews*: Web sites such as *rottentomatoes.com* and *epinions.com* offer their users a forum in which to solicit feedback and reviews of various movies and products. Moreover, some large consumer web sites such as *Amazon* and *BestBuy* offer the same capabilities for their users for an enormous quantity of products. As a result, consumers are now faced with another problem. Reading reviews from other users is very time-consuming, and many reviews are often in conflicts. SA allows us to automatically analyze such review data, learn from it, and make useful predictions.
- *Sentiment search*: Apart from augmenting the capabilities of existing systems, SA can also be integrated into a general-purpose search engine. Currently, when searching for products or movies in any commonly used search engine such as Google or Bing, the results consist of descriptions, company websites and related items. If the user is searching for reviews on a product or movie, she will undoubtedly have to navigate the results further. However, if the search engine knows that the user is searching for a review, it could bring up relevant review results closer to the top, and rank them according to polarities. It could even use SA techniques to automatically generate a summary or score based on the relevant information that it finds.
- *Marketing and business intelligence*: Not only does SA have an ability to quickly aggregate, organize and summarize all kinds of user reviews and present the results to other consumers, it can also be used in the other direction to add value to the producers. In order for companies to remain competitive, they invariably must research the tastes of their customers very carefully. Instead of spending a lot of time sifting through customer reviews of their products, companies can use SA techniques to quickly aggregate and organize reviews into information that can help them learn about the relationship between their customers and their products. Another important area that companies have to consider when marketing their products is to target their products to consumers where the probability of conversion is the highest. Since there are many blogs that utilize contextual ad systems such as Google AdWords, the connection between a blog post about a product and the product advertisement should be as relevant as possible. For example, it does not make sense for a product to be advertised on a blog in which the review of that particular product is negative. By the same token, if a blog is reviewing a product favorably, the ads should target that product as much as possible to increase the likely conversion rate.
- *Detection of inflammatory text and cyber-bullying*: Perhaps an application area that is not immediately obvious for SA is to detect overly heated or antagonistic language. However, since at its heart, SA deals with opinions about products, places, things, it can readily deal with opinions about other people. Inappropriate remarks about individuals can be analyzed and eventually removed using SA techniques. As social media websites such as Facebook and Twitter are gaining more and more popularity, especially with children, cyber-bullying will only be a greater problem. Parents are rightly concerned that their children's online presence could be misused and slandered by their peers. SA techniques could provide valuable methods to help combat this recent phenomenon.

1.2. General challenges

Traditional text classification seeks to classify a document by topic and often relies on word frequencies for making a decision. For example, if a document has a high frequency of certain key words like '*football*', there is an increased probability that the document is about *sports*. However, since SA deals with opinions about topics and not topics themselves, the classification approaches are necessarily different from those for the traditional (or topical) text classification.

First of all, the indicators of sentiment must be aligned to the language that a reviewer is most likely to use when writing a review. Early work [4,5] has focused on using adjectives such as '*good*' and '*bad*' and adverbs like '*terrifically*' and '*hatefully*' as the important indicators of sentiment. Intuitively, this is what we would expect of an opinionated document. However, later research [13,11,7,18] also suggests that other parts of speech such as verbs and even nouns [14] could be valuable indicators of sentiment.

The second difference between SA and topical text classification is how the indicators are combined to infer the polar category (either positive or negative) of a document. Pang et al. [13] shows that using the frequencies of the indicators (as it is done in topical text classification) has a negative impact. Therefore, they propose a scheme based on *presence* (whether an indicator occurs in the document or not). Moreover, it seems that subjective indicators occur less frequently than topical indicators [19], thus re-affirming the presence-based approach.

Thirdly, opinion and subjectivity are quite domain dependant. This is a little counter-intuitive because we tend to describe a wide array of things using the same words. For example, a movie can be described as '*great*', and so can a product. However, there are examples where the same word offers different meanings in different domains. One such example is the word: '*unpredictable*' [17]. It has positive connotations when referring to a movie, but negative ones when dealing with a product.

Finally, since user-generated documents are written using free-form natural language and tend to be rather informal with only liberal use of proper spelling and grammar, the task of somehow extracting the salient information, representing it and finally making predictions based on it is ultimately challenging.

1.3. Motivations

As illustrated by the earlier example, the task of classifying a review document can be explored by taking into account the differences between entities and their modifiers. Specifically, we can think of a review document as a mixture of topical words, referred to as *entities*, that do not carry any sentiment, and modifying words that relate a certain entity with a polarity. An important characteristic of review documents is that the reviewers tend to discuss the whole set of entities throughout the entire document, whereas the modifiers for those entities tend to be more localized at the sentence or phrase level. In other words, each entity can be *polymorphous* within the document, with a long-range *semantic* relationship between its forms while the modifiers in each case are bound to the entity in a short-range, *syntactic* relationship. Generalizing a single entity to all the entities that are found in a document, and taking all their respective modifiers into account, we can start to infer the polarity of the entire document based on the set of all the modifiers. This reduces to finding all the syntactic words in the document and disregarding the entities.

Taking another look at the modifiers that appear in the earlier example, we might assume that all of the relevant indicators for SA come from specific parts of speech categories such as *adjectives* and *adverbs*, while other parts of speech classes such as nouns are more relevant for general text classification, and can be discarded. However, as demonstrated by Pang et al. [13], Pang and Lee [11], Hu and Liu [7], and Riloff et al. [14], there are some nouns and verbs that are useful sentiment indicators as well. Therefore, a clear distinction cannot be made along parts of speech categories.

To address this issue, we propose a *feature selection* scheme in which we can obtain important sentiment indicators that:

1. Do not rely on specific parts of speech classes while maintaining the focus on syntax words.
2. Separate semantic words that do not indicate sentiment while keeping nouns that do.
3. Reflect the domain for the set of documents.

With feature selection schemes that focus on the outlined sentiment indicators as a basis for our machine learning approach, we should achieve competitive accuracy results when classifying document polarities. More specifically, we propose a set of new feature selection schemes that use a Content and Syntax model [3] to automatically learn a set of features in a review document by separating the entities that are being reviewed from the subjective expressions that describe those entities in terms of polarities. Then, we focus only on the subjective expressions while ignoring the entities so that we can choose more salient features for document-level sentiment analysis.

The rest of this article is organized as follows. In Section 2, we discuss some important work and results for SA and outline the modeling and classification techniques used by our approach. In Section 3, we provide details about our feature selection methods. Our experiments and analyses are given in Section 4, and finally, conclusions and future directions are presented in Section 4.5.

2. Related work

2.1. Feature selection in sentiment analysis

The majority of the approaches for SA involve a two-step process:

1. Identify the parts of the document that will likely contribute to *positive* or *negative* sentiments.
2. Combine these parts of the document in ways that increase the odds of the document falling into one of these two polar categories.

Many statistical feature selection methods for topical text classification can also be used for SA. The simplest statistical approach for feature selection by Pang et al. [13] is to use the most frequently-occurring words in the corpus as polarity indicators. This approach is commonly used with general text classification, and the results achieved indicate that simple document frequency cutoffs can be an effective feature selection scheme.

The main advantage of statistical approaches for feature selection is that they are fully automatic. However, they often fail to separate features that carry sentiment from those that do not. This is because words that are used frequently are not guaranteed to be subjective. In fact, it is usually the topical words that are repeated more often. Moreover, according to Pang and Lee [12], most reviews tend to use many different words for sentiment. Therefore, such approaches can pick up on topical words that do not carry any subjectivity.

The most common approach, used by researchers such as Das and Chen [1], starts with a manually created lexicon specific to their particular domain whereas others [8,21] attempt to craft a general-purpose opinion lexicon that can be used across domains. More recent lexicon-based approaches [2,7,9,14] begin with a small set of 'seed' words and bootstrap this set through synonym detection or various on-line resources to obtain a larger lexicon.

However, lexicon-based approaches have several key difficulties. First, they take time to compile. Whitelaw et al. [18] report that their feature selection process took 20 person-hours, since it involves work done by human annotators. In separate qualitative experiments done by Pang et al. [13], Wilson et al. [20] and Kim and Hovy [9], the agreement between human judges when given a list of sentiment-bearing words is as low as 58% and no higher than 76%. In addition, some words may not be frequent enough for a classification algorithm. Clearly, a feature selection method that utilizes the advantages of both approaches while minimizing their disadvantages is desired.

2.2. Topic modeling and HMM-LDA

Topic models such as *Latent Dirichlet Allocation* (LDA) are generative models that allow documents to be explained by unobserved (latent) topics. The Hidden Markov Model LDA (HMM-LDA) [3] is a topic model that simultaneously models topics and syntactic structures in a collection of documents. The idea behind the model is that a typical word can play different roles. It can either be part of the content and serve in a semantic (topical) purpose or it can be used as part of the grammatical (syntactic) structure. It can also be used in both contexts.

HMM-LDA models this behavior by inducing syntactic classes for each word based on how they appear together in a sentence using a Hidden Markov Model. Each word gets assigned to a syntactic class, but one class is reserved for the semantic words. Words in this class behave as they would in a regular LDA topic model, participating in different topics and having certain probabilities of appearing in a document. More formally, the model is defined in terms of three sets of variables and a *generative process*. Let $\mathbf{w}=\{w_1,\ldots,w_n\}$ be a sequence of words where each word w_i is one of V words; $\mathbf{z}=\{z_1,\ldots,z_n\}$, a sequence of topic assignments where each z_i is one of K topics; and $\mathbf{c}=\{c_1,\ldots,c_n\}$, a sequence of class assignments where each c_i is one of C classes. One class, $c_i=1$ is designated as the 'semantic class', and the rest, the 'syntactic' classes.

Since we are dealing with a Hidden Markov Model, we require a variable representing the *transition probabilities* between the classes, given by a $C\times C$ *transition matrix* π that models transitions between classes c_{i-1} and c_i and is drawn from $Dir(\gamma)$. The generative process for document d is described as follows:

1. Sample topic proportions $\theta^{(d)}$ from a Dirichlet prior $Dir(\alpha)$
2. For each word w_i in document d:
   (a) draw $z_i\sim\theta^{(d)}$
   (b) draw $c_i\sim\pi^{(c_i-1)}$
   (c) if $c_i=1$, then draw $w_i\sim\phi^{(z_i)}$, else draw $w_i\sim\phi^{(c_i)}$

where $\phi^{(z_i)}\sim Dir(\beta)$ and $\phi^{(c_i)}\sim Dir(\delta)$, both from *Dirichlet* distributions.

As with the basic LDA model, the exact inference for HMM-LDA is intractable. However, inferences of this model can be done via Gibbs sampling where we iteratively draw a topic assignment z_i and a class assignment c_i for each word w_i in the corpus. Once the inference is done, we can then use the class assignments to distinguish between syntactic words and semantic words in a review document.

2.3. Text classification based on maximum entropy modeling

Maximum entropy modeling [10] is a framework whereby the features represent constraints on the overall model and the idea is to incorporate the knowledge that we have while preserving as much uncertainty as possible about the knowledge we do not have. The features f_i are binary functions where there is a vector x representing input elements (unigram features in our case) and c, the class label for one of the possible categories. More specifically, a feature function is defined as follows:

$$f_{i,c'}(x,c)=\begin{cases}1 & \text{if } x \text{ contains } w_i \text{ and } c=c'\\ 0 & \text{otherwise}\end{cases} \quad (1)$$

where word w_i and category c' correspond to a specific feature.

Employing the feature functions described above, a Maximum Entropy model takes the following form:

$$P(x,c)=\frac{1}{Z}\prod_{i=1}^{K}\alpha_i^{f_i(x,c)} \quad (2)$$

where K is the number of features, α_i is the weight for feature f_i, and Z is a normalizing constant. By taking the logarithm on both sides, we get the log-linear model:

$$\log P(x,c)=-\log Z+\sum_{i=1}^{K}f_i(x,c)\log\alpha_i. \quad (3)$$

To classify a document, we compute $P(c|x)$ so that the c with the highest probability will be the category for the given document.

To learn the parameters for a maximum entropy model, we need to obey the following constraints, each of which expresses a characteristic of the training data that should also be present in the learned distribution:

$$Ep^* f_i = E\tilde{p} f_i \tag{4}$$

for a maximum entropy model distribution p^* and the *empirical* distribution $\tilde{p}$. The empirical distribution is calculated directly from the training data, while the expectation of the currently estimated model is approximated as follows:

$$Epf_i = \sum_{x,c} \tilde{p}(x)p(c|x)f_i(x,c) = \frac{1}{N}\sum_{j=1}^{N}\sum_{c} p\left(c\middle|x_j\right)f_i\left(x_j,c\right) \tag{5}$$

where $c \in C$ is a category.

There are two *iterative optimization* algorithms that provide initial settings for model parameters and repeatedly refine these parameters to bring them closer to an optimal solution. The first algorithm is called *Generalized Iterative Scaling (GIS)*, which requires that the sum of the features for each possible combination (x,c) be equal to a constant C, which is defined as the greatest possible feature sum over *all possible data*. The *Improved Iterative Scaling* (IIS) algorithm is a variation of GIS that does not require such a condition.

Convergence can be attained by setting a maximum number of iterations prior to training, or when the sum of the differences in the values of the parameters between iterations is below a certain threshold.

3. Feature selection (FS) based on HMM-LDA

3.1. Characteristics of salient features

To motivate our approach, we first describe criteria that are useful in selecting salient features for SA:

1. *Features should be expressive enough to add useful information to the classification process*. As discussed in Section 1, the most expressive features in terms of polarity are the *modifying* words that describe an entity in a certain way. These are usually, but not restricted to, adjectives, adverbs, subjective verbs and nouns.
2. *All features together should form a broad and comprehensive viewpoint of the entire corpus*. In a corpus of many documents, some features can represent a subset of the corpus very accurately, while other features may represent another subset of the corpus. The problem arises when representing the whole corpus with a specific feature set [15].
3. *Features should be as domain-dependent as possible*. Examples from Hurst and Nigam [8] and Das and Chen [1] as well as many other approaches indicate that SA is a domain-dependant task, and the final features should reflect the domain of the corpus that they are representing.
4. *Features must be frequent enough*. Rare features do not occur in many documents and make it difficult to train a machine learning algorithm. Experiments by Pang et al. [13] indicate that having more features does not help learning, and the best accuracy was achieved by selecting a subset of features based on *document frequency*.
5. *Features should be discriminative enough*. A learning system needs to be able to pick up on their presence in certain documents for one outcome and absence in other documents for another outcome in classification.

3.2. FS based on syntactic classes

Our proposed FS scheme is to utilize HMM-LDA to obtain words that, for the most part, follow the criteria we set out in Section 3.1. We train an HMM-LDA model to give us the syntactic classes that we further combine to form our final features. Let word $w_i \in V$ where V is the vocabulary. Also let $c_j \in C$ be a class. We define $P_{c_j}(w_i)$ as the probability of word w_i in class c_j, and one class, $c_j = 1$ indicates the semantic class. Since each class (syntactic and semantic) has a probability distribution over all words, we need to select words that offer a good *representation* of the class. The representative words in each class have a much higher probability than the other words. Therefore, we can select the representative words by the *cumulative probability*. Specifically, we select the top percentage of the words in a class whereby the sum of their probabilities will be within some pre-defined range. This is necessary since there are many words in each class with low probabilities in which we are not interested [16]. The cumulative distribution function is defined as:

$$F_j(w_i) = \sum_{P_{c_j}(w) \geq P_{c_j}(w_i)} P_{c_j}(w) \tag{6}$$

Then, we can define the set of words in class c_j as:

$$W_{c_j} = \left\{ w_i \middle| F_j(w_i) \leq \eta \right\} \tag{7}$$

where η is a pre-defined threshold such that $0 \leq \eta \leq 1$. Next, we define the set of words in all the syntactic classes W_{syn} as:

$$W_{syn} = \left\{ w_i \middle| w_i \in W_{c_j} \text{ and } c_j \neq 1 \right\} \tag{8}$$

and the set of words in the semantic class W_{sem} as:

$$W_{sem} = \left\{ w_i \middle| w_i \in W_{c_j} \text{ and } c_j = 1 \right\} \tag{9}$$

Since modifying words for sentiment typically fall into syntactic classes, we could use words in W_{syn} as features for SA. However, as observed by Pang et al. [13], the best classification performance is achieved by a subset of features (typically around 2500). As a general step, we can apply a document frequency (DF) cutoff to select the most frequent features. Let $df(w_i)$ denote the document frequency of word w_i, indicating the number of documents in which w_i occurs in the corpus. Then the resulting features selected based on df can be defined as:

$$cut\left(W_{syn}, \epsilon\right) = \left\{ w_i \middle| w_i \in W_{syn} \text{ and } df(w_i) \geq \epsilon \right\} \tag{10}$$

where is the minimum document frequency required for feature selection.

3.3. FS based on set difference between syntactic and semantic classes

The main characteristic of using HMM-LDA classes for feature selection is that the set of words in the syntactic classes and the set of words in the semantic class are not disjoint. In fact, there is quite a large overlap. In this and the next subsections, we discuss ways to remedy and even exploit this situation to get a higher level of accuracy. In the Pang et al. movie review data, there is about 35% overlap between words in the syntactic and semantic classes for $\eta = 0.9$. Our first systematic approach attempts to gain better accuracy by lowering the ratio of semantic words in the final feature set.

More formally, given the set of syntactic words W_{syn}, we can reduce the overlap with W_{sem} by doing a set difference operation:

$$W_{syn} - W_{sem} \tag{11}$$

This will give us all the words that are more favored in the syntactic classes. However, as we shall see shortly, and also as we earlier speculated, by subtracting all the words in the semantic class, we are actually getting rid of some useful features. This is because (a) it

is possible for the semantic class to contain words that are syntactic, and as a result are useful, and (b) there exist some semantic words that are good indicators of polarity. Therefore, we seek to 'lessen' the influence of the semantic class by cutting only a certain portion of the words in it, but not all of them.

For the above scheme, we outline Algorithm 1 that enables us to select features from W_{syn} by applying a percentage cutoff for W_{sem} and then doing a set difference operation. We define $top(W_{sem}, \delta)$ to be the δ% of the words with top probabilities in W_{sem}.

Algorithm 1: Syntactic–Semantic Set Difference
Data: W_{syn} and W_{sem} as input
1 $W'_{sem} = top(W_{sem}, \delta)$
2 $W_{diff} = W_{syn} - W'_{sem}$
3 $W'_{syn} = cut(W_{diff}, \epsilon)$

Note that when $\delta = 1.0$, we get the same result as $W_{syn} - W_{sem}$. In our experiments, we try a range of δ values for SA.

3.4. FS based on max scores of syntactic features

The running theme through the HMM-LDA feature selection schemes is that if a word is highly ranked (has a high probability of occurring) in a syntactic class, we should use that word in our feature set. Moreover, if a word is highly ranked in the semantic class, we usually do not want to use that word in our feature set because the word usually indicates a frequent noun. Therefore, the desirable words are those that occur with high probability in the syntactic classes, but do not occur with high probability in the semantic class, or do not occur there at all.

To this end, we have formulated a scheme that adds such words to our feature set. For each word, we obtain its highest probability in the set of syntactic classes. Comparing this probability with the probability of the same word in the semantic class, we disregard the word if the probability in the semantic class is greater.

We define the *max scores* for word w_i for both the syntactic and semantic classes and describe how we select features based on the max scores in Algorithm 2

Algorithm 2. Max scores of syntactic features

Data: $c_j \in C$ where $1 \leq j \leq |C|$
foreach $w_i \in V$ **do**
  $S_{syn}(w_i) = max_{c_j \neq 1} P_{c_j}(w_i)$
  $S_{sem}(w_i) = P_{c_1}(w_i)$
  $W_{max} = \{w_i | S_{syn}(w_i) > S_{sem}(w_i)\}$
$W'_{syn} = cut(W_{max}, \epsilon)$

4. Experiments

This section describes the steps taken to generate some experimental results for each scheme described in the previous section. Before we can analyze these sets of results, we take a look at some baselines.

4.1. Dataset and evaluation

We use the corpus of 2000 movie reviews [11] that consists of 1000 positive and 1000 negative documents selected from on-line forums. General statistics for the corpus are summarized in Table 1, where *average word length* is the total number of characters divided by the total number of words in the corpus; *average sentence length* is the total number of words divided by the total number of sentences; *average document length* is the total number of words divided by the total number of documents; and *lexical diversity score* is the average number of times each word type appears in a document. The table also shows the standard deviations for the related statistics.

Table 1
General statistics for the movie review corpus.

Corpus totals		
Total number of word tokens	1,583,820	
Total number of word types	39,764	
	Average	Std. dev.
Word length	4.32	0.468
Sentence length	23.13	5.680
Document length	791.91	347.251
Lexical diversity score	1.79	0.445

As can be seen, the corpus does have a high document length variance. In fact, the shortest document has 19 words whereas the longest has 2879 words. The Lexical Diversity score suggests that on average, there is not much word repetition, which is in line with the observation that reviewers tend to use a wider range of language to describe an opinion [12].

In our experiments, we randomize the documents and split the data into 1800 for training/testing purposes and 200 as the validation set. For the 1800 documents, we run a 3-fold cross validation procedure where we train on 1200 documents and test on 600. We compare the resultant feature sets after each FS scheme using the OpenNLP[2] Maximum Entropy classifier.

For all the classification tasks, we start with a particular FS scheme to rank and select features. After that, we further choose a set of 2500 final features for classification based on the document frequency cutoff method. In addition, all maximum entropy models are trained with the IIS algorithm without smoothing.

Throughout these experiments, we are interested in the *classification accuracy*. This is evaluated simply by comparing the resultant class from the classifier and the actual class annotated by Pang and Lee [11]. The number of matches is divided by the number of documents in the *test* set. Thus, given an *annotated* test set $d_{set_A} = \{(d_1, o_1), (d_2, o_2), \ldots (d_S, o_S)\}$ and the classified set, $d_{set_B} = \{(d_1, q_1), (d_2, q_2), \ldots (d_S, q_S)\}$, we calculate the accuracy as follows:

$$\frac{\sum_{i=1}^{S} I(o_i = q_i)}{S} \tag{12}$$

where $I(\cdot)$ is the indicator function.

4.2. Baseline results

After replicating the results from Pang et al. [13], we varied the number of iterations per fold by using a held-out validation set 'eval'. The higher accuracy achieved suggests that the model was not fully trained after 10 iterations (as used in Pang et al. [13]).

In order to compare with our HMM-LDA based schemes, we ran experiments to explore a basic POS-based feature selection scheme. In this approach, we first tagged the words in each document with POS tags and selected the most frequently-occurring unigrams that were not tagged as 'NN', 'NNP', 'NNS' or 'NNPS' (the 'noun' categories). This corresponds to *POS (-NN*)* in Table 2.

Next, we tagged all the words and only selected the words that were tagged as 'JJ*', 'RB*', and 'VB*' categories (the 'syntactic' categories). The idea is to include as part of the feature set all the words that are not

[2] http://incubator.apache.org/opennlp/.

Table 2
Baseline results with a different number of iterations. Each column represents a different feature selection method.

Iterations	DF cutoff	POS (-NN*)	POS (JJ* + RB* + VB*)
10	0.821	0.827	0.811
25	0.836	0.831	0.824
eval	0.845	0.848	0.826

'semantically oriented'. This corresponds to *POS (JJ* + RB* + VB*)* in Table 2.

4.3. HMM-LDA training

Our feature selection methods involve training an HMM-LDA model on Pang et al.'s [13] corpus of movie reviews, taking the class assignments, and combining the resultant unigrams to create features for the MaxEnt classifier. Since HMM-LDA is an *unsupervised* topic model, we can train it on the entire corpus. We trained the model using the Topic Modeling Toolbox[3] MATLAB package on the 2000 movie reviews. Since the HMM-LDA model requires sentences to be outlined, we used the usual end-of-sentence markers ('.', '!', '?', ':'). The training parameters are **T = 50** topics, **S = 20** classes, **ALPHA = 1.0**, **BETA = DELTA = 0.01**, and **GAMMA = 0.1**. We found that 1000 iterations is sufficient as we tracked the log-likelihood of every 10 iterations. After training, we have both the topic assignments **z** and the class assignments **c** for each word in all the samples.

To demonstrate how the HMM-LDA classes are assigned to words, we randomly select a movie review document and show a paragraph of the text marked with different classes by our trained model.

...Of course, it helps that Kaye^{0} has an actor^{16} as talented^{14} as Norton^{0} to play this part^{4}. It's astonishing how frightening^{14} Norton^{0} looks with ashaved head and a swastika on his chest. In addition to getting the look just right, he^{13}'s perfect^{6} for this role – Derek^{0} requires intelligence, depth, and a whole lot of shouting, and Norton^{0} does it all with ease. Even when he^{13}'s at his meanest, Derek^{0} has a likable^{14} quality to him, and that's a gutsy approach when telling a story about a skinhead. What adds depth to the story is a subplot in which the principal^{16} of Danny's school (AveryBrooks^{0,0}) becomes obsessed with purging the hatred from Danny^{0}. The other performances^{2} are all terrific^{14}, with standouts from Furlong^{0}, D'Angelo^{0}, and Lien^{0}. Visually, the film^{16} is very powerful^{14}. Kaye^{0} indulges in a lot of interesting^{14} artistic^{14} choices, and most of them^{4} work^{11} nicely^{4} –

The first thing to notice about the paragraph above is that all the names such as 'Kaye' and 'Norton' are in class 0 (the semantic class). The second thing to notice is that some of the adjectives such as 'talented' and 'frightening' are in the same class. Another result worth mentioning is that the assigned classes do not necessarily follow parts-of-speech categories. This is in line with the expected premise that HMM-LDA is a more fine-grained approach, as discussed in Section 1. However, we note that in this example, some common nouns such as 'actor', 'film' and 'principal' are in their own class, and not necessarily in class 0.

4.4. Selecting features based on syntactic classes

In this experiment we fix $\eta = 0.9$ to get the top words in each class having a cumulative probability of 90%. These are the *representative* words in each class which we merge into W_{syn}. Finally, we select 2500 words by the DF cutoff method. This list of words is then used as features for the MaxEnt classifier. We run the classifier for 10, 25 and 'eval' number of iterations in order to compare with the baseline results.

At $\eta = 0.9$, there are 6189 words in W_{syn} before we select the top 2500 using the *df* cutoff. From Table 3, we see that the accuracy has increased from 0.845 to 0.863 at the 'eval' number of iterations.

In all of our experiments, we use *df* cutoff to get a manageable number of features for the classifier. This is partly based on Pang et al. [13] and partly based on calculating the *Pearson correlation* for each class between the document frequency and word probability at $\eta = 0.9$. Since every class has a positive correlation in the range of [0.313938, 0.888160] where the average is 0.576, we can say that there is a correlation between the two values.

4.5. Selecting features based on set difference

The result for set difference is derived by varying the percentage of top semantic words that should be excluded in the final feature set. For example, some words in $W_{syn} \cap W_{sem}$ that have a higher probability in W_{sem} are: '*hollywood*', '*war*', and '*fiction*' while the words that have a higher probability in W_{syn} include: '*good*', '*love*' and '*funny*'. The δ value is defined by the percentage of the words in W_{sem} that we exclude from W_{syn}. The results for $0.0 \le \delta \le 1.0$ for increments of $\delta \times |W_{sem}|$, are summarized in Table 4.

From the results, we can see that as we remove more and more words from W_{sem}, the accuracy level decreases. This suggests that $W_{sem} \cap W_{syn}$ contains some important features and if we subtract W_{sem} entirely, we essentially eliminate them. At each cutoff level, we are eliminating 10% until we have eliminated the whole set. Clearly, a more fine-grained approach is needed, and that leads us to the Max-Score results.

4.6. Selecting features based on max scores

For the method based on max scores, we may select features that are in both W_{sem} and W_{syn} sets as long as their max scores in W_{syn} are higher than those in W_{sem}.

Comparing the accuracy in Table 5 with those in the previous subsections, we can say that using the fine-grained Max-Score algorithm improves the classification accuracy. This means that iteratively removing words that have a relatively higher probability in W_{sem} compared to W_{syn} does not eliminate important words occurring in both sets, but lessens the influence of some high probability words in W_{sem}.

4.7. Discussion of the results

For our experiments, the best accuracy is achieved by utilizing the Max-Score algorithm (outlined in Section 4.4) after a further selection of 2500 with the *df* cutoff. As discussed in Section 4.4, the Max-Score

Table 3
Results for FS based on syntactic classes at 10, 25 and 'eval' iterations.

Iterations	FS BASED on syntactic features
10	0.823
25	0.839
eval	0.863

[3] http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm.

Table 4
Results for FS based on the Syntactic–Semantic set difference method. Each row represents the accuracy achieved at a particular δ value.

δ	FS based on set difference	δ	FS based on set difference
0.0	0.861	0.5	0.852
0.1	0.862	0.6	0.846
0.2	0.865	0.7	0.849
0.3	0.858	0.8	0.847
0.4	0.857	0.9	0.840
	1.0	0.831	

algorithm enables us to select words that have a higher score in W_{syn} than in W_{sem}. This approach has the dual advantage of keeping the words that are present in both W_{syn} and W_{sem} but have higher scores in W_{syn} and ignoring the words that are also present in both sets but have higher scores in W_{sem}. Ultimately, this decreases the influence of the frequent and overlapped words that have a high probability in W_{sem}.

Finally, to quantify the significance level of our best approach against the baseline methods in Section 4.2, we calculated the p-values for the one-tailed t-tests comparing our best approach based on max scores with the DF and POS (-NN*) baselines, respectively. The resulting p-values of 0.011 and 0.014 suggest that our best approach is *significantly* better than the baseline approaches.

5. Conclusions and future directions

We approached the task of Feature Selection for Sentiment Analysis by using a Content and Syntax model, known as HMM-LDA, to separate the *entities* in a review document from the potentially sentiment carrying *modifiers*. HMM-LDA models entities and modifiers as long-range and short-range dependencies, respectively, allowing us to separate words into semantic and syntactic classes. By grouping all the sentiment modifiers for the entities in a document into the syntactic classes, we are selecting the features that are intuitively in line with the outlined characteristics of salient features for SA (see Section 4.1).

The proposed feature selection schemes achieved competitive results in our experiments for document polarity classification. Using only the syntactic classes (which correspond to the modifiers), we can immediately see an improvement in the classification accuracy for review documents. By reducing the overlaps with the semantic words in our final feature sets, we were able to continue improving the classification performance. In particular, the fine-grained approach based on the max scores showed significant improvements in the classification accuracy over the baseline approaches, as well as the approach based solely on syntactic classes. Essentially, this improvement in accuracy can be attributed to the removal of some of the non-sentiment-carrying features from the semantic class and replacing them with more appropriate sentiment-carrying features from the syntactic classes.

Overall, the increase in accuracy over the baseline approaches in all of our feature selection schemes can be explained by the discriminative nature of the modifying words that our schemes pick up. One of the main characteristics for salient features is that they must be discriminative enough so that the learning system is able to pick up on their presence in certain documents relating to one outcome and the absence in other documents relating to another outcome. Since the HMM-LDA syntactic classes by and large pick up modifying words that describe entities, these words by definition cannot be both positive and negative at the same time, given the domain. It is then important to minimize the impact of *neutral* (non-sentiment-carrying) features. In all our approaches, we minimize the impact by separating the semantic class from the syntactic classes, and as a result, removing some of the neutral features that are present in the baseline schemes.

Table 5
Result for FS based on max scores.

Iterations	FS based on max scores
eval	0.875

We have thus far only experimented on our feature selection schemes with the movie review domain, since the data are readily available, preprocessed, and annotated. Moreover, according to Pang and Lee [12], the accuracy levels achieved for movie reviews are generally lower than those for product reviews and hotel reviews. This is due to the homogeneous nature of movie reviews where reviewers evaluate the characteristics of the entities in movies using more expressive language. It will be interesting to test our approach for other domains. For example, to evaluate our feature selection schemes on product reviews, we would have to train an HMM-LDA model on that specific domain and use a maximum entropy classifier to evaluate the sentiment classification accuracy as we have done for the movie reviews.

Another avenue for future development of this framework could include identifying and extracting *aspects* from a review document. So far, we have not identified aspects from the entities, choosing instead to classify a document as a whole. However, this framework can be readily applied to extract relevant (most probable) aspects using the LDA topic model and then restrict the syntactic modifiers to the range of sentences where an aspect occurs. This would give us an *unsupervised* aspect extraction scheme that we can combine with a classifier to predict polarities for each aspect.

Finally, we can extend our framework to classify review documents based on a *scale*. Thus far we have focused on the *binary* classification task. A review document can be either 'positive' or 'negative'. However, many applications need a further level of detail such as 'how positive' and 'how negative'. By adding a further level of categorization in the form of a scale (for example 1 to 5, where 1 is 'very bad' to 5 being 'very good'), we can use the same features that we have gathered from the HMM-LDA model but simply add more categories for the maximum entropy classifier. Of course, we would need the necessary annotated data to accomplish this task and the accuracy levels may be reduced simply because the classifier would have more categories to handle.

Acknowledgment

The authors would like to acknowledge the financial support from Ontario Centres of Excellence (OCE) through the OCE/Precarn Alliance Program.

References

[1] Sanjiv R. Das, Mike Y. Chen, Yahoo! for Amazon: sentiment extraction from small talk on the Web, Management Science 53 (9) (2007) 1375–1388.
[2] Xiaowen Ding, Bing Liu, Philip S. Yu, A holistic lexicon-based approach to opinion mining, Proceedings of the Conference on Web Search and Web Data Mining (WSDM), 2008.
[3] Thomas L. Griffiths, Mark Steyvers, David M. Blei, Joshua B. Tenenbaum, Integrating topics and syntax, In Advances in Neural Information Processing Systems, 17, MIT Press, 2005, pp. 537–544.
[4] Vasileios Hatzivassiloglou, Kathleen McKeown, Predicting the semantic orientation of adjectives, Proceedings of the Joint ACL/EACL Conference, 1997, pp. 174–181.
[5] Vasileios Hatzivassiloglou, Janyce Wiebe, Effects of adjective orientation and gradability on sentence subjectivity, Proceedings of the International Conference on Computational Linguistics (COLING), 2000.
[6] John A. Horrigan, Online shopping, Pew Internet & American Life Project Report, 2008.
[7] Minqing Hu, Bing Liu, Mining opinion features in customer reviews, Proceedings of AAAI, 2004, pp. 755–760.
[8] Matthew Hurst, Kamal Nigam, Retrieving topical sentiments from online document collections, Document Recognition and Retrieval XI, 2004, pp. 27–34.
[9] Soo-Min Kim, Eduard Hovy, Determining the sentiment of opinions, Proceedings of the International Conference on Computational Linguistics (COLING), 2004.
[10] Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, USA, 1999.
[11] Bo Pang, Lillian Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, Proceedings of the Association for Computational Linguistics (ACL), 2004, pp. 271–278.
[12] Bo Pang, Lillian Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (January 2008) 1–135.

[13] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79–86.
[14] Ellen Riloff, Janyce Wiebe, Theresa Wilson, Learning subjective nouns using extraction pattern bootstrapping, Proceedings of the Conference on Natural Language Learning (CoNLL), 2003, pp. 25–32.
[15] Fabrizio Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (1) (2002) 1–47.
[16] Mark Steyvers, Tom Griffiths, Probabilistic topic models, in: T. Landauer, D. Mcnamara, S. Dennis, W. Kintsch (Eds.), Latent Semantic Analysis: A Road to Meaning, Laurence Erlbaum, 2006.
[17] Peter Turney, Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, Proceedings of the Association for Computational Linguistics (ACL), 2002, pp. 417–424.
[18] Casey Whitelaw, Navendu Garg, Shlomo Argamon, Using appraisal groups for sentiment analysis, Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM), ACM, 2005, pp. 625–631.
[19] Janyce Wiebe, Learning subjective adjectives from corpora, Proceedings of AAAI, 2000.
[20] Theresa Wilson, Janyce Wiebe, Paul Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), 2005, pp. 347–354.
[21] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, Wayne Niblack, Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques, Proceedings of the IEEE International Conference on Data Mining (ICDM), 2003.

Adnan Duric is a recent M.Sc. graduate in Computer Science from the University of Guelph, Canada. His primary interests include areas of Natural Language Processing such as Information Retrieval, Sentiment Analysis and Topic Modeling. Other interests include large-scale data analysis, regression and optimization.

Fei Song is an Associate Professor at the School of Computer Science, University of Guelph, Canada. His research interests are mainly in the areas of Statistical Natural Language Processing, including Information Retrieval, Text Categorization, Text Segmentation, Sentiment Analysis, and Text Summarization. He also worked and collaborated with Industrial Companies in the last ten years and has applied various techniques of Natural Language Processing to real world problems. He received his Ph.D. in Computer Science from the University of Waterloo, Canada, and soon after graduation, joined the University of Guelph as a faculty member.